Parallel HDF5

Quincey Koziol
koziol@hdfgroup.org
The HDF Group

What is HDF?

- HDF stands for Hierarchical Data Format
- A file format for managing any kind of data
- Software system to manage data in the format

- Designed for high volume or complex data
- Designed for every size and type of system
- Open format and software library, tools

- There are two HDF’s: HDF4 and HDF5
- Today we focus on HDF5

Parallel HDF5 Currently

- MPI for communication and I/O
  - Application passes in a communicator to duplicate and use for opening the file and exchanging information
- Metadata Create/Modify/Delete
  - Must be collective ☹
- Raw data I/O
  - Collective and independent I/O supported
  - But, compressed dataset writes not supported

Parallel HDF5 Future

- New DOE Funding:
  - “ExaHDF5” – Project w/LBNL & PNL to enhance HDF5 and aim for exascale platforms
  - “Scalable HDF5” – Contract w/LLNL to enhance HDF5 and explore high-performance, non-MPI-I/O solutions
  - “Damsel” – Project w/ANL, NWU & ORNL to design and implement a next generation file format and I/O middleware package

ExaHDF5 Tasks

- Remove “collective” restriction for metadata modifications
  - Including supporting compressed datasets
- Add metadata and raw data indexing to HDF5
- Add support for asynchronous parallel I/O
- Design and implement file system autotuning mechanism
- Support “ordered updates” in parallel

Single-Writer/Multiple-Reader Access

- Situation: A long-running process is modifying an HDF5 file and simultaneously other processes want to inspect data in the file.
- Solution: Single-Writer/Multiple-Reader (SWMR) File Access, using “ordered updates”
  - Allows simultaneous reading of HDF5 file while the file is being modified by another process
  - No inter-process coordination necessary
- Bonus! Crash-proofs file also! ☺

Scalable HDF5 Tasks

- Explore and implement alternate scalable I/O approaches:
  - “Poor man’s parallel I/O” (PMPIO) (from LLNL)
  - “Reduced-Blocking I/O” (rbIO) (from ANL)
- Design new Virtual File Drivers tuned for “modern” parallel file systems
- Metadata aggregation & alignment in file
- Advanced page buffering within library
- Deferred/staged/segregated object creation

Other Planned HDF5 Tasks

- Design and implement “Virtual Object Layer” within HDF5
  - Allows creation of plugins operating at higher-level of abstraction that Virtual File Layer
  - HDF5 data model, without using HDF5 file
  - Can we merge HDF5 with [parallel] file system?
- Expand HDF5 data model
  - Support “shared” dataspaces
  - Attributes on datatypes and dataspaces (allows units on datatypes, etc.)
- “Append-only” library and file format optimizations

Parallel HDF5 Challenges

- We are implementing file system *on top of* MPI-I/O!
  - Not enough support in MPI for necessary locking operations, etc.
  - Difficult to create production-quality software in a portable and cost-effective way
- Need more funding
  - Support and reach out to HPC application development teams
  - Keep up with research efforts: ADIOS, pnetCDF, etc.